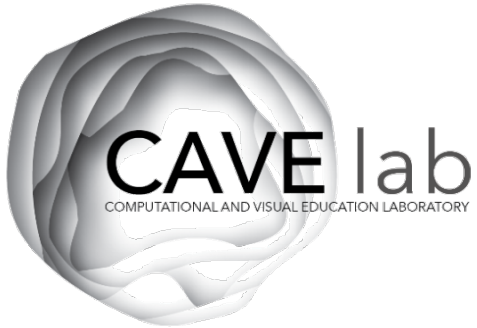


MIT CAVE lab Hackathon

Welcome & Introductions

Matthias Winkenbach, PhD
MIT Center for Transportation & Logistics
Director, MIT Computational and Visual Education (CAVE) Lab
Director, MIT Megacity Logistics Lab

MIT Center for Transportation & Logistics

Agenda

Saturday, February 10, 2018

10:00 – 11:00	Registration & Refreshments
11:00 – 11:30	**Welcome & Introductions to the hackathon**
11:30 – 13:00	**“Team speed dating”, lunch, team registration**
13:00	**Hacking begins**
15:00 – 18:00	*Optional: hourly short CAVE lab tours*

Agenda

Sunday, February 11, 2018

09:00 – 12:00	*Optional: hourly short CAVE lab tours*
14:30	**Hacking ends, submissions due**
14:30 – 15:30	**Relax, refresh and prepare voting pitches**
15:30 – 19:15	**Pitches & Voting**
19:15 – 20:00	***Award Ceremony***
20:00	*Adjourn*

Team Speed Dating

Saturday, February 10, 2018

11:30 – 13:00 **“Team Speed dating” & lunch**

11:30 – 11:45 Round 1
11:45 – 12:00 Round 2
12:00 – 12:15 Round 3

12:15 – 12:45 Team registration

"Speed dating" rules

1. If you already have a team, disregard.

2. Form trios and introduce yourself
   - your skills
   - your interests
   - why you're here
   - your ideas for the hackathon

3. If happy, you may remove yourself from the speed dating.

4. Otherwise, repeat step 3.

3 rounds, 15 minutes per round, 45 minutes in total

Data & Instructions

1. Visit and create an account at **cave-hack.devpost.com**

2. On the day of the event, download the **data and instructions** on the main page

3. At the end of the hackathon, use the Devpost **submission** function to submit your work
   - Link to GitHub repository
   - Link to running public demo
   - A short description of your work, it's features and capabilities

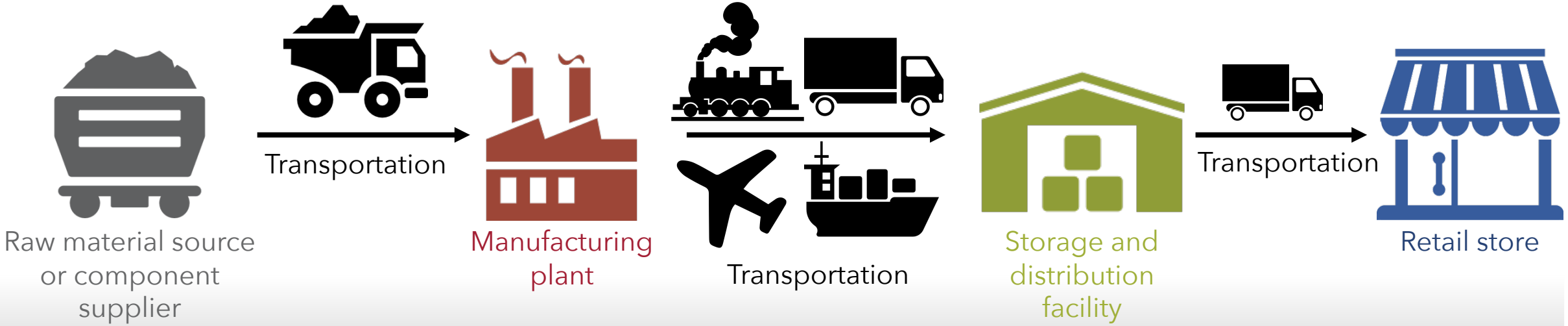
   more details on submissions will be shared in a few minutes

The objective of the hackathon: a visual analytics application

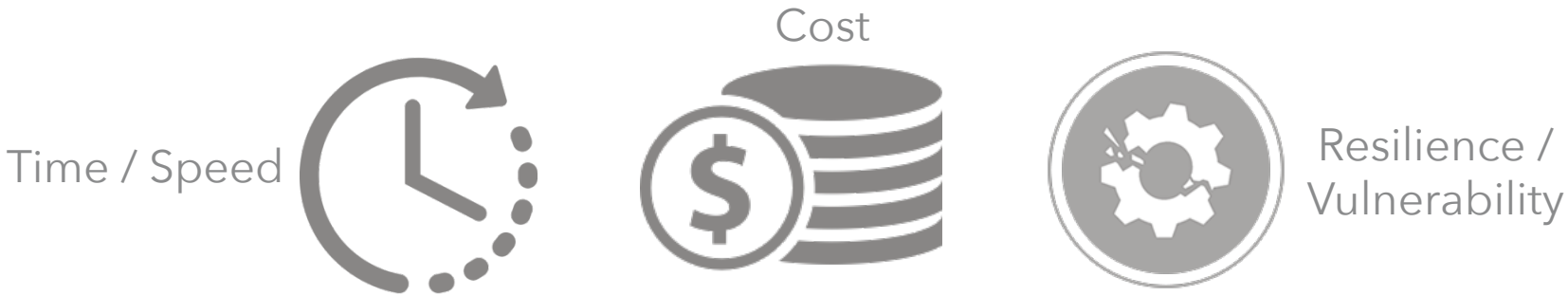
Participants are asked use object oriented programming (JavaScript, Python, etc.) to develop a **visually appealing**, **touch-interactive**, **browser-based** application that enables humans to intuitively **visualize, explore, understand, analyze, and manipulate** large and complex networks.

The data set you will be working on contains information from real-world, global supply networks

Context: Global supply networks

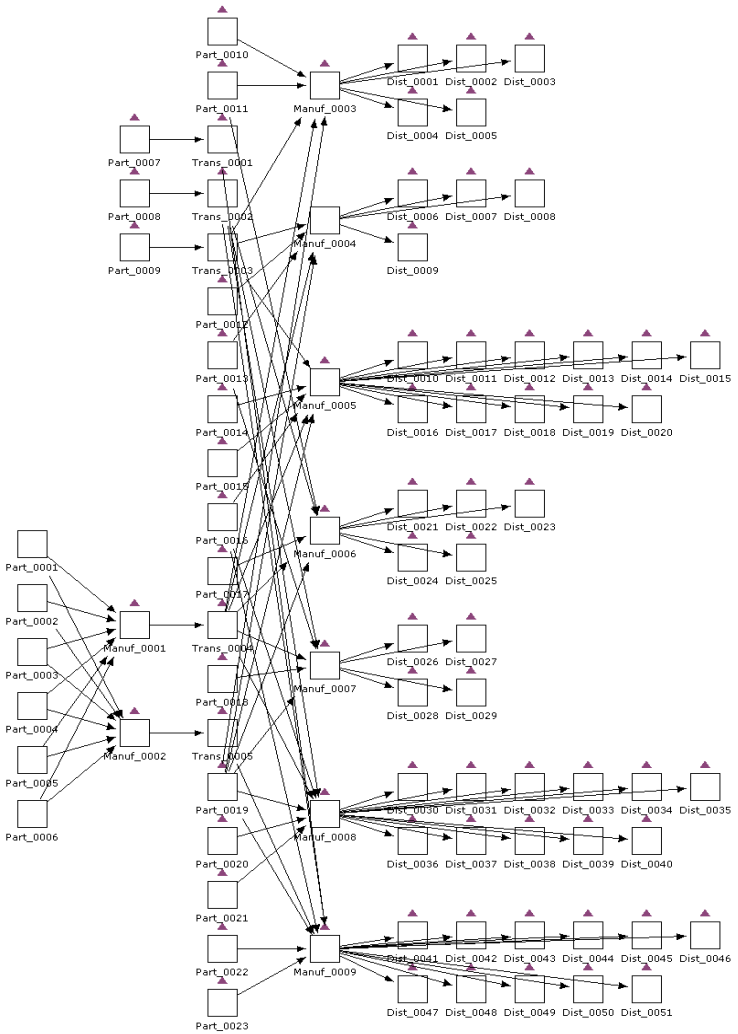


Performance Metrics & Properties



Supply networks consist of various types of stages (i.e. nodes) connected by (directed) arcs

Example: Supply Network Map of Cereal Co.



Types of Stages

1. **Part_** The **source** of a basic product component or raw material (e.g. a mine, or a components supplier)

2. **Manuf_** A **manufacturing facility** that creates / changes a product or puts multiple product components together to form a product

3. **Dist_** A **distribution facility** at which a product or component is packaged and stored (e.g. a warehouse or distribution center)

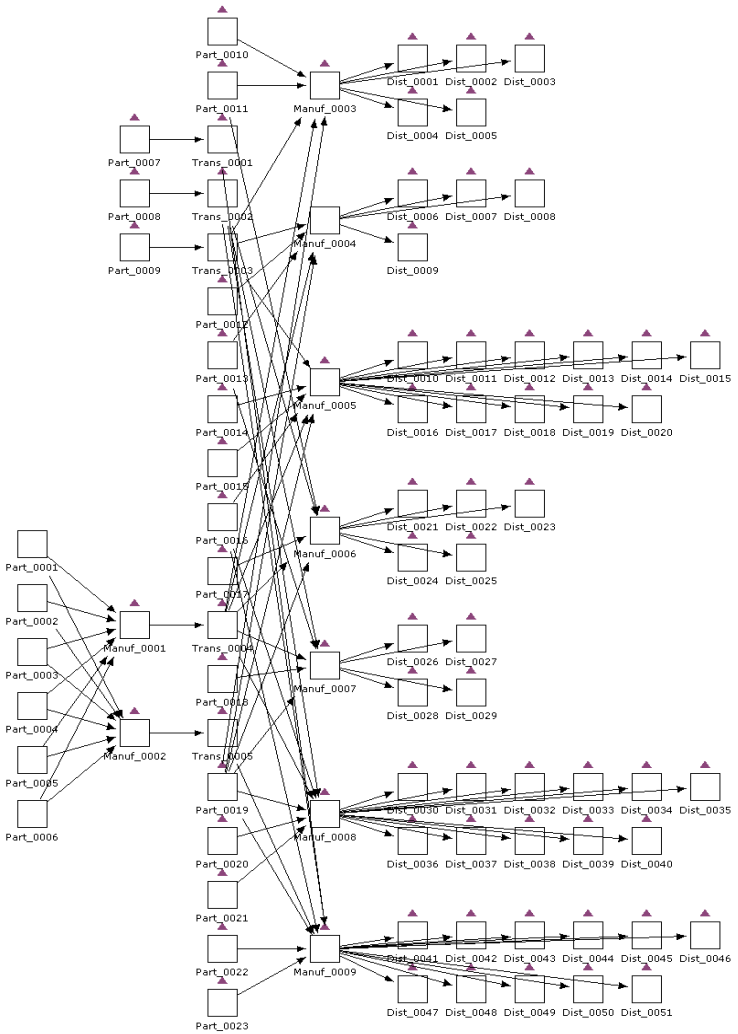
4. **Retail_** A **retail establishment** at which the product is stored and presented to the consumer (e.g. a store, shop, restaurant, etc.)

5. **Transp_** A transportation activity, moving a product or component from one stages to another stage (e.g. a trucking company, express delivery company, etc.)

© 2018 Dr. Matthias Winkenbach | MIT Computational and Visual Education Lab – chart no. 8

The demand for units of a product occurring at certain nodes defines the flows through the network

Example: Supply Network Map of Cereal Co.



Typical Stage Sequences

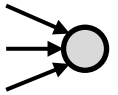
1. **Part_ → Transp_ → Manuf_ → Transp_ → Dist_ → Transp_ → Retail_**
2. **Manuf_ → Transp_ → Dist_ → Transp_ → Retail_**
3. **Part_ → Transp_ → Manuf_ → Transp_ → Dist_**

…or combinations thereof.

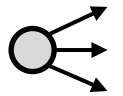
Note: sometimes, there is no transportation stage between the other stages. This means there is no transportation requirement between the two. For example, this may be the case if the raw material source (a Part_ stage) or the warehouse (a Dist_ stage) is directly adjacent to the manufactuing plant (a Manuf_ stage).

Demand vs. Supply Nodes

Demand node A node that only has inflowing arcs, but no outflowing arcs. Here, the product leaves the supply network:
a) bought by a commercial customer: usually at Dist_ stages
b) bought by a private consumer: usually at Retail_ stages



Supply node A node that only has outflowing arcs, but no inflowing arcs. Here, the product or its components enter the supply network for the first time: usually at Part_ or Manuf_ stages



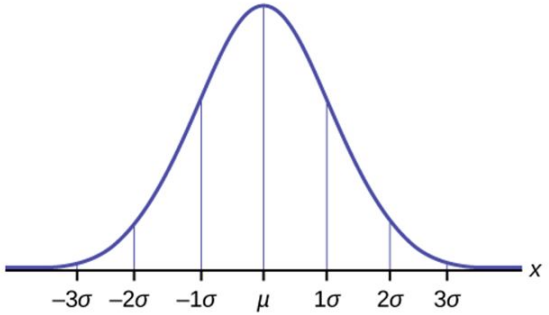
© 2018 Dr. Matthias Winkenbach | MIT Computational and Visual Education Lab – chart no. 9

The time it takes for a unit to pass through the stages of a network may be deterministic or probabilistic

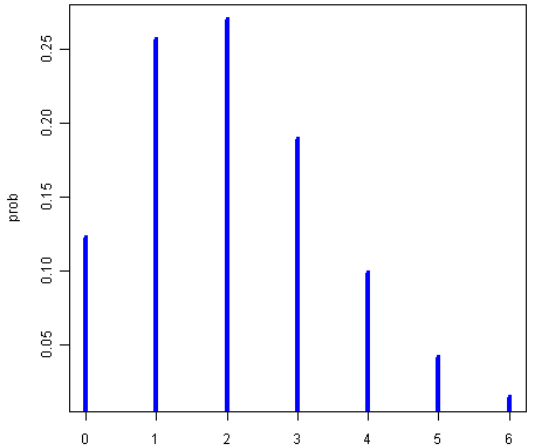
The stage times can be defined in three possible ways:

1. Stage times are **deterministic** – they always take the same value, for every unit passing through the stage.

2. Stage times are assumed to be **normally distributed** – with a mean stage time and a standard deviation the stage time being given by the data set.



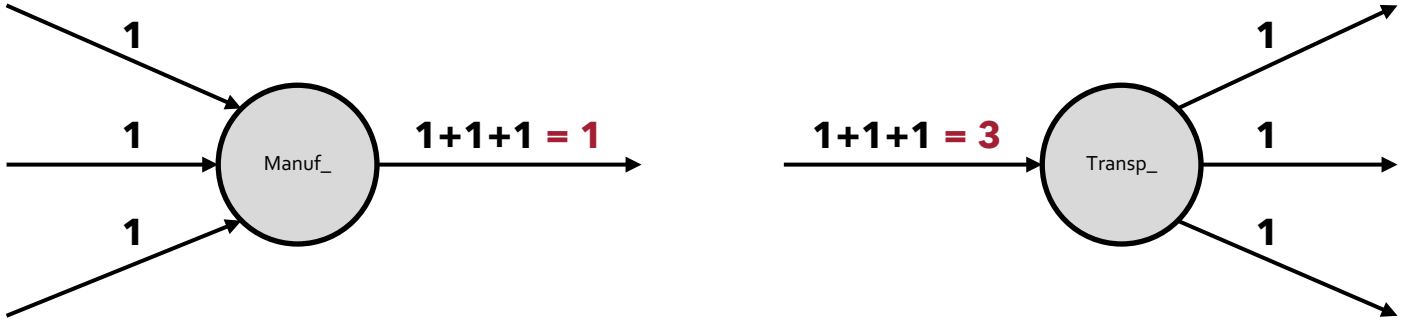
3. Stage times are assumed to follow a **discrete probability distribution** – with up to six possible time realizations and their respective likelihoods being given by the data set.



© 2018 Dr. Matthias Winkenbach | MIT Computational and Visual Education Lab – chart no. 10

A few more notes on demands and flows of units through the networks

1. At each stage that is neither a demand nor a supply stage, **one unit of output of that stage requires one unit of input** from each of the stages that feed into that stage through inflowing arcs.



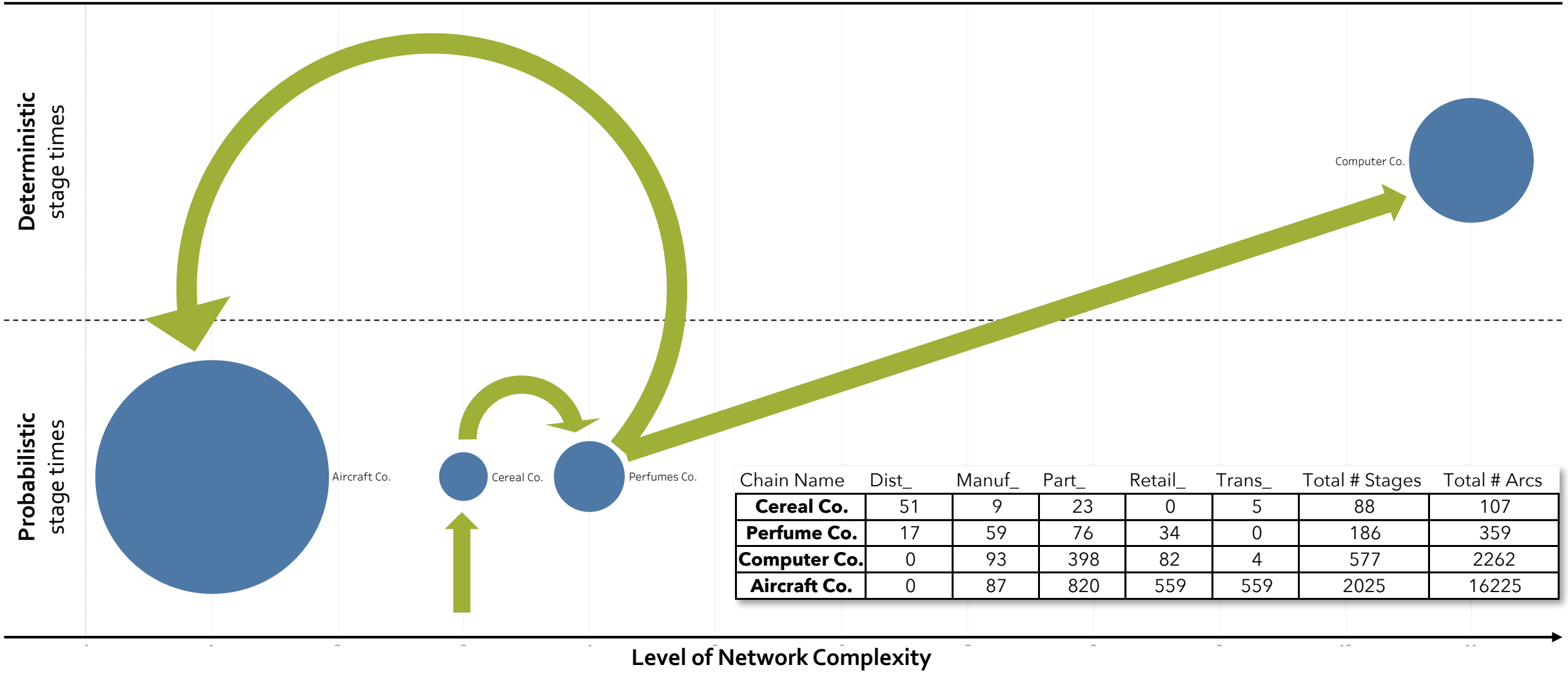
2. The (normally distributed) **demands at the demand stages** (i.e., nodes without any outflowing arcs) **propagate back** to non-demand nodes as independent and identically distributed (i.i.d.) random variables. That means:
   - The demands and their probability distributions are mutually independent.
   - As a demand from a demand stage propagates through the network, its probability distribution does not change.

For every stage (i.e., node of the network), you are provided with a variety of information

Stage Data (i.e. node properties)

Variable name	Meaning	Explanation	Unit
Stage Name	Unique stage identifier	-	-
stageCost	Direct cost added to the total product cost at this stage.	-	US Dollars
relDepth	Relative depth index at this stage	The net number of forward arcs minus backward arcs traversed between the current stage and a root node in the network that is assigned a relative depth of zero.	-
stageClassification	Type of stage	A stage can be a Part_, Manuf_, Dist_, Retail_, or Transp_ stage.	-
avgDemand	Average daily demand occurring at the stage	Only defined for demand stages (no outflowing arcs). Demands are assumed to be normally distributed around this mean.	units
stdDevDemand	Standard deviation of daily demand occuring at the stage.	Only defined for demand stages (no outflowing arcs). Demands are assumed to be normally distributed with this standard deviation.	units
stageTime	Processing time of each unit going through this stage.	If stageTime is probabilistic, this is the mean.	days
stdDev_stageTime	Standard debviation of the processing time per unit at this stage.	Only defined if stageTime is probabilistic.	days
stageTime_1 .. 6	Discrete possible realizations of stageTime	Only defined if stageTime is characterized by a discrete distribution. Otherwise, stageTime is assumed to be normally distributed.	days
stageTime_%_1 .. 6	Probability of discrete realization of stageTime	Only defined if stageTime is characterized by a discrete distribution. Otherwise, stageTime is assumed to be normally distributed.	-

The supply networks in the dataset have different characteristics – we suggest: start simple, then expand!



Chain Name	Dist_	Manuf_	Part_	Retail_	Trans_	Total # Stages	Total # Arcs
Cereal Co.	51	9	23	0	5	88	107
Perfume Co.	17	59	76	34	0	186	359
Computer Co.	0	93	398	82	4	577	2262
Aircraft Co.	0	87	820	559	559	2025	16225

© 2018 Dr. Matthias Winkenbach | MIT Computational and Visual Education Lab – chart no. 13

Your work will be evaluated along three major dimensions: visualization, interaction, analytics

Visualization

- How effective is your application in communicating the structural properties, characteristics, and performance of large and complex network structures?
- How does your application support a human in intuitively distinguishing the important from the less important features of the network?
- Which complementary visual representations on the data are you using to provide a better understanding of the network and its features?

We suggest: have a look at what you find online on different methods of graph and network visualization

Interaction

- How does your application support a natural, intuitive exploration, analysis, and manipulation of the network structure?
- How does your application highlight the important network features and provide more detail on specific aspects / parts of the network on demand?
- How intuitive is the interaction with you user interface?

We suggest: test your application on a tablet or iPad to improve your interactions

Analytics

- How effective is your application in describing the structural features and performance of the networks? Which metrics / statistics are you using?
- How insightful are the predictive analyses (e.g., „how will the network perform under certain circumstances?“) and/or prescriptive analyses (e.g., „how should the network be modified to become more robust?“) you perform on the networks?

We suggest: look into graph theory and network science methods for metrics and algorithms that may be of interest

You can choose your own emphasis on the three dimensions of visual analytics

All three dimensions go hand in hand and your applications should cover all of them to some extend.

You can put equal weight on all three dimensions, or you can choose to put more emphasis on some of them, and less emphasis on others.

Please state and breifly motivate your choice when submitting your hack.

Optional Lab Tours

Tour times

- Saturday, February 10:
  3 pm – 6 pm
- Sunday, February 11:
  9 am – 12 pm

How to take part

- Tours start at every full hour
- Meeting point: in front of the elevators, 3rd floor
- Duration approx. 20 min
- Max. 10 people per tour

Submissions

- You will develop a **touch-interactive** and **browser-based** interface
- You must deploy your demo to a **public URL**
  - You are free to use whatever means necessary
  - We have documented some free options that are sufficient (Heroku, Azure, GitHub Pages, Firebase)
- Use **GitHub** for your code repository
- Complete your submission at **cave-hack.devpost.com** by the deadline:

no later than **2:30 PM on Sunday**

- Let us know, if you need help or have any questions
- You can find all details at cave-hack.devpost.com

Submissions

- Enter your submission at **cave-hack.devpost.com**
- **Only one submission per team** – team members can edit the same submission
- Prefix your project title with your **assigned team number** – Ex: **"0: Network Visualization Hack"**
- You should include:
  - The relevance of **visualization, interactivity, analytics** to your project
  - **A link to your GitHub repository**
  - **A link to a working demo of your application**
- Other things to include:
  - Your motivations
  - How does it work
  - How you built it
  - What you have learned
  - Any challenges you ran into

Pitches & Voting

Sunday, February 11, 2018

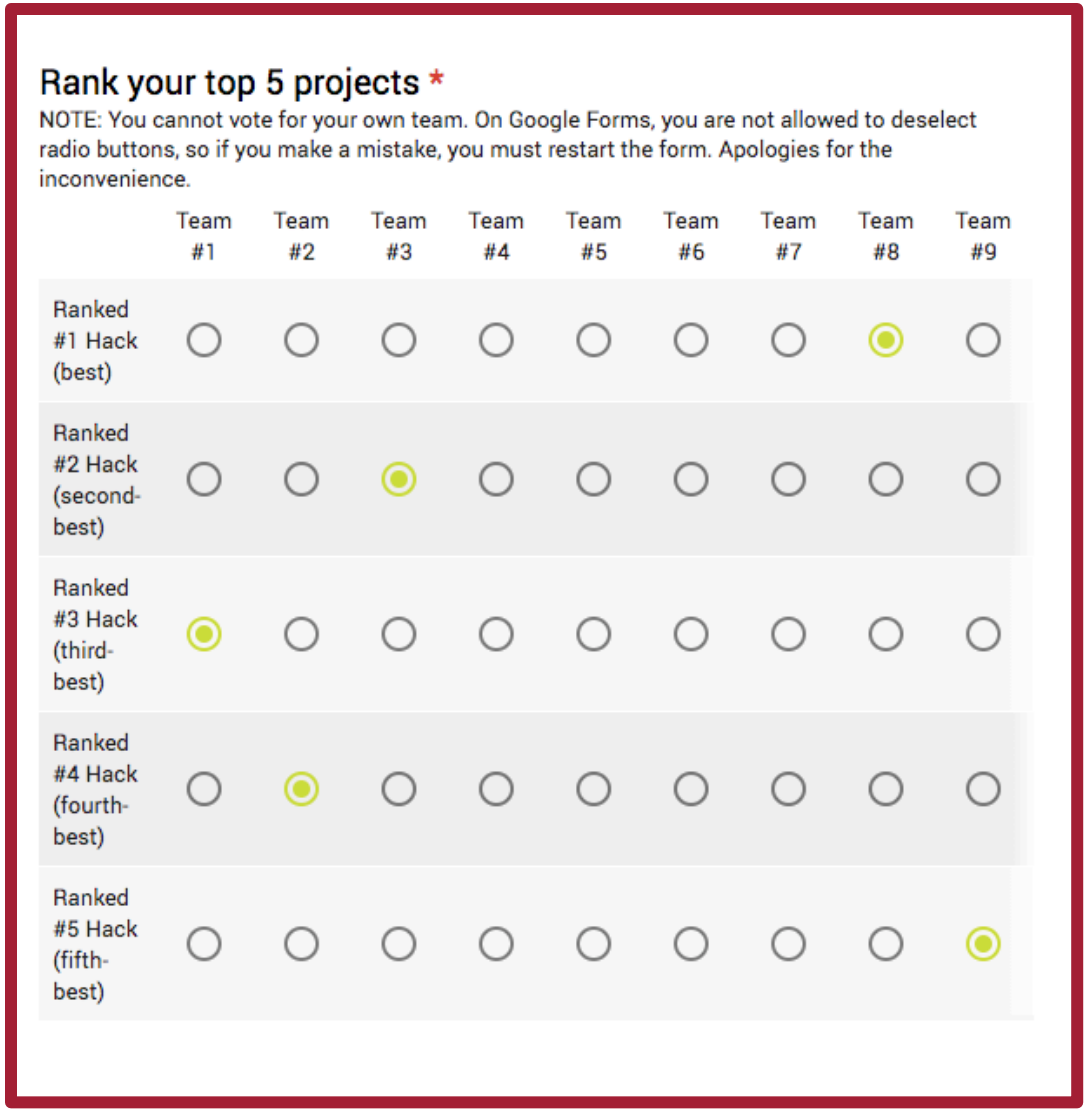
15:30 – 19:15 **Pitches & Voting**

15:30 – 16:30	*Group 1*
16:30 – 16:45	*Refreshment break and group changeover*
16:45 – 17:45	*Group 2*
17:45 – 18:00	*Refreshment break and group changeover*
18:00 – 19:00	*Group 3*
19:00 – 19:15	*Refreshment break, determination of winners*

Every team will be assigned to a group. When it's your group's turn, you'll be presenting your hack to the other participants. If it's the other groups' turn, you'll go around the room, and listen to the pitches of the others.

After all three rounds, you have to cast your votes.

How to Vote



Rank your top 5 projects *

NOTE: You cannot vote for your own team. On Google Forms, you are not allowed to deselect radio buttons, so if you make a mistake, you must restart the form. Apologies for the inconvenience.

	Team #1	Team #2	Team #3	Team #4	Team #5	Team #6	Team #7	Team #8	Team #9
Ranked #1 Hack (best)	○	○	○	○	○	○	○	◉	○
Ranked #2 Hack (second-best)	○	○	◉	○	○	○	○	○	○
Ranked #3 Hack (third-best)	◉	○	○	○	○	○	○	○	○
Ranked #4 Hack (fourth-best)	○	◉	○	○	○	○	○	○	○
Ranked #5 Hack (fifth-best)	○	○	○	○	○	○	○	○	◉

Voting Rules

1. You can vote for <u>your top 5 teams</u>.
2. You <u>cannot vote for your own team</u>. If you do, your vote won't count.
3. In case you vote multiple times, only your <u>most recent entry</u> will count.
4. If you don't vote, your team won't be eligible for the awards.

We will be monitoring the votes as they come in. If your submission is invalid or missing, we will contact you via email.

Food Schedule

Saturday, February 10, 2018

Lunch	11:30 – 13:00
Dinner	18:00 – 19:00

Sunday, February 11, 2018

00:00 – 01:00	**Midnight Snack**
07:00 – 08:00	**Breakfast**
12:00 – 13:00	**Lunch**

Rules and Procedures

Our hackathon is dedicated to providing a **harassment-free** experience for everyone, regardless of gender identity and expression, age, sexual orientation, disability, physical appearance, body size, race, ethnicity, nationality, religion, coding experience, etc. Sexual language and imagery is not appropriate at this hackathon in any form.

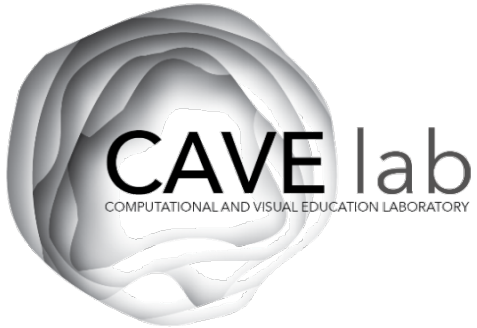
We do not tolerate harassment of hackathon participants in any form. Note that all hacks are also subject to the same anti-harassment policies.

If taking **photos** of the event, please allow other hackathon participants the opportunity to opt out of the photograph. It is inappropriate to take photographs of unwilling subjects, or in places where **privacy** is expected (i.e. bathrooms).

Hackathon participants violating these rules may be expelled at the discretion of the hackathon staff.

Whom to Contact:

- Hackathon staff at the front desk:
  - cave-door@mit.edu
  - 617-715-4206
- If the hackathon staff cannot help you, please contact Matthias Winkenbach:
  - cavelab@mit.edu
  - 857-253-1639 (please ONLY use for emergencies)



Thank you.

Questions?

cavelab@mit.edu